Matematický software

Zápočtový dokument

**Jméno:** Karel Pinka

**Kontaktní email:** Charles.pinka@gmail.com​

**Datum odevzdání:** 10.7.2023

**Odkaz na repozitář:** https://github.com/Charlesv98?tab=repositories

# Formální požadavky

**Cíl předmětu:**

Cílem předmětu je ovládnout vybrané moduly a jejich metody pro jazyk Python, které vám mohou být užitečné jak v dalších semestrech vašeho studia, závěrečné práci (semestrální, bakalářské) nebo technické a výzkumné praxi.

**Získání zápočtu:**

Pro získání zápočtu je nutné částečně ovládnout alespoň polovinu z probraných témat. To prokážete vyřešením vybraných úkolů. V tomto dokumentu naleznete celkem 10 zadání, která odpovídají probíraným tématům. Vyberte si 5 zadání, vypracujte je a odevzdejte. Pokud bude všech 5 prací korektně vypracováno, pak získáváte zápočet. Pokud si nejste jisti korektností vypracování konkrétního zadání, pak je doporučeno vypracovat více zadání a budou se započítávat také, pokud budou korektně vypracované.

**Korektnost vypracovaného zadání:**

Konkrétní zadání je považováno za korektně zpracované, pokud splňuje tyto kritéria:

1. Použili jste numerický modul pro vypracování zadání místo obyčejného pythonu
2. Kód neobsahuje syntaktické chyby a je interpretovatelný (spustitelný)
3. Kód je čistý (vygooglete termín clean code) s tím, že je akceptovatelné mít ho rozdělen do Jupyter notebook buněk (s tímhle clean code nepočítá)

**Forma odevzdání:**

Výsledný produkt odevzdáte ve dvou podobách:

1. Zápočtový dokument
2. Repozitář s kódem

Zápočtový dokument (vyplněný tento dokument, který čtete) bude v PDF formátu. V řešení úloh uveďte důležité fragmenty kódu a grafy/obrázky/textový výpis pro ověření funkčnosti. Stačí tedy uvést jen ty fragmenty kódu, které přispívají k jádru řešení zadání. Kód nahrajte na veřejně přístupný repozitář (github, gitlab) a uveďte v práci na něj odkaz v titulní straně dokumentu. Strukturujte repozitář tak, aby bylo pro nás hodnotitele intuitivní se vyznat v souborech (doporučuji každou úlohu dát zvlášť do adresáře).

**Podezření na plagiátorství:**

Při podezření na plagiátorství (významná podoba myšlenek a kódu, která je za hranicí pravděpodobnosti shody dvou lidí) budete vyzváni k fyzickému dostavení se na zápočet do prostor univerzity, kde dojde k vysvětlení podezřelých partií, nebo vykonání zápočtového testu na místě z matematického softwaru v jazyce Python.

**Kontakt:**

Při nejasnostech ohledně zadání nebo formě odevzdání se můžete obrátit nejlépe na tvůrce těchto zadání na Discord serveru Pavla Beránka (měli byste mít odkaz) nebo na emailu: *pavelberanek91@gmail.com*.

# 1. Knihovny a moduly pro matematické výpočty

**Zadání:**

V tomto kurzu jste se učili s některými vybranými knihovnami. Některé sloužily pro rychlé vektorové operace jako numpy, některé mají naprogramovány symbolické manipulace, které lze převést na numerické reprezentace (sympy), některé mají v sobě funkce pro numerickou integraci (scipy). Některé slouží i pro rychlé základní operace s čísly (numba).

Vaším úkolem je změřit potřebný čas pro vyřešení nějakého problému (např.: provést skalární součin, vypočítat určitý integrál) pomocí standardního pythonu a pomocí specializované knihovny. Toto měření proveďte alespoň pro 5 různých úloh (ne pouze jiná čísla, ale úplně jiné téma) a minimálně porovnejte rychlost jednoho modulu se standardním pythonem. Ideálně proveďte porovnání ještě s dalším modulem a snažte se, ať je kód ve standardním pythonu napsán efektivně. ​

**Řešení:**

První úloha se zabývá výpočtem faktoriálu čísla 100. Standardní Python využívá modul math a jeho funkce factorial, zatímco numpy má vlastní implementaci této funkce. Měří se čas potřebný pro výpočet oběma způsoby.

Druhá úloha se zaměřuje na výpočet průměru listu čísel. Standardní Python využívá vestavěnou funkci sum a dělení, zatímco numpy poskytuje funkci mean, která toto provádí efektivněji. Měří se čas potřebný pro výpočet oběma způsoby.

Třetí úloha se týká výpočtu determinantu matice 3x3. Opět se porovnává standardní Python s numpy. V obou případech se používá funkce det z modulu numpy.linalg.

Čtvrtá úloha se zaměřuje na výpočet skalárního součinu dvou vektorů. Standardní Python využívá komprehenzi seznamu a následný součet, zatímco numpy poskytuje funkci dot, která je efektivnější. Měří se čas potřebný pro výpočet oběma způsoby.

Pátá úloha se zabývá výpočtem určitého integrálu. Standardní Python používá numerickou metodu Riemannova integrálu s pevným počtem dílků a sumačním vzorcem. Na druhé straně používá knihovna scipy funkci quad pro numerickou integraci pomocí adaptivní metody quadpack. Měří se čas potřebný pro výpočet oběma způsoby.

Pro každou úlohu je měřen čas pomocí funkce time.time() před a po provedení výpočtu, a následně je výsledek vypsán spolu s naměřeným časem.

# 2. Vizualizace dat

**Zadání:**

V jednom ze cvičení jste probírali práci s moduly pro vizualizaci dat. Mezi nejznámější moduly patří matplotlib (a jeho nadstavby jako seaborn), pillow, opencv, aj. Vyberte si nějakou zajímavou datovou sadu na webovém portále Kaggle a proveďte datovou analýzu datové sady. Využijte k tomu různé typy grafů a interpretujte je (minimálně alespoň 5 zajímavých grafů)​. Příklad interpretace: z datové sady pro počasí vyplynulo z liniového grafu, že v létě je vyšší rozptyl mezi minimální a maximální hodnotou teploty. Z jiného grafu vyplývá, že v létě je vyšší průměrná vlhkost vzduchu. Důvodem vyššího rozptylu může být absorpce záření vzduchem, který má v létě vyšší tepelnou kapacitu.

**Řešení:**

doplňte

# 3. Úvod do lineární algebry

**Zadání:**

Důležitou částí studia na přírodovědecké fakultě je podobor matematiky zvaný lineární algebra. Poznatky tohoto oboru jsou základem pro oblasti jako zpracování obrazu, strojové učení nebo návrh mechanických soustav s definovanou stabilitou. Základní úlohou v lineární algebře je nalezení neznámých v soustavě lineárních rovnic. Na hodinách jste byli obeznámeni s přímou a iterační metodou pro řešení soustav lineárních rovnic. Vaším úkolem je vytvořit graf, kde na ose x bude velikost čtvercové matice a na ose y průměrný čas potřebný k nalezení uspokojivého řešení. Cílem je nalézt takovou velikost matice, od které je výhodnější využít iterační metodu.

**Řešení:**

Kód vytváří graf, který porovnává průměrný čas řešení soustav lineárních rovnic pomocí přímé a iterační metody pro různé velikosti čtvercových matic. Cílem je najít velikost matice, při které je výhodnější použít iterační metodu. K tomu se generují náhodné matice, měří se časy řešení a vytváří se graf závislosti času na velikosti matice. Tímto způsobem lze identifikovat optimální velikost matice pro efektivní použití iterační metody.

# 

# 4. Interpolace a aproximace funkce jedné proměnné

**Zadání:**

Během měření v laboratoři získáte diskrétní sadu dat. Často potřebujete data i mezi těmito diskrétními hodnotami a to takové, které by nejpřesněji odpovídaly reálnému naměření. Proto je důležité využít vhodnou interpolační metodu. Cílem tohoto zadání je vybrat si 3 rozdílné funkce (např. polynom, harmonická funkce, logaritmus), přidat do nich šum (trošku je v každém z bodů rozkmitejte), a vyberte náhodně některé body. Poté proveďte interpolaci nebo aproximaci funkce pomocí alespoň 3 rozdílných metod a porovnejte, jak jsou přesné. Přesnost porovnáte s daty, které měly původně vyjít. Vhodnou metrikou pro porovnání přesnosti je součet čtverců (rozptylů), které vzniknou ze směrodatné odchylky mezi odhadnutou hodnotou a skutečnou hodnotou.

**Řešení:**

doplňte

# 5. Hledání kořenů rovnice

**Zadání:**

Vyhledávání hodnot, při kterých dosáhne zkoumaný signál vybrané hodnoty je důležitou součástí analýzy časových řad. Pro tento účel existuje spousta zajímavých metod. Jeden typ metod se nazývá ohraničené (například metoda půlení intervalu), při kterých je zaručeno nalezení kořenu avšak metody typicky konvergují pomalu. Druhý typ metod se nazývá neohraničené, které konvergují rychle, avšak svojí povahou nemusí nalézt řešení (metody využívající derivace). Vaším úkolem je vybrat tři různorodé funkce (například polynomiální, exponenciální/logaritmickou, harmonickou se směrnicí, aj.), které mají alespoň jeden kořen a nalézt ho jednou uzavřenou a jednou otevřenou metodou. Porovnejte časovou náročnost nalezení kořene a přesnost nalezení.

**Řešení:**

doplňte

# 6. Generování náhodných čísel a testování generátorů

**Zadání:**

Tento úkol bude poněkud kreativnější charakteru. Vaším úkolem je vytvořit vlastní generátor semínka do pseudonáhodných algoritmů. Jazyk Python umí sbírat přes ovladače hardwarových zařízení různá fyzická a fyzikální data. Můžete i sbírat data z historie prohlížeče, snímání pohybu myší, vyzvání uživatele zadat náhodné úhozy do klávesnice a jiná unikátní data uživatelů.

**Řešení:**

doplňte

# 7. Metoda Monte Carlo

**Zadání:**

Metoda Monte Carlo představuje rodinu metod a filosofický přístup k modelování jevů, který využívá vzorkování prostoru (například prostor čísel na herní kostce, které mohou padnout) pomocí pseudonáhodného generátoru čísel. Jelikož se jedná spíše o filosofii řešení problému, tak využití je téměř neomezené. Na hodinách jste viděli několik aplikací (optimalizace portfolia aktiv, řešení Monty Hall problému, integrace funkce, aj.). Nalezněte nějaký zajímavý problém, který nebyl na hodině řešen, a získejte o jeho řešení informace pomocí metody Monte Carlo. Můžete využít kódy ze sešitu z hodin, ale kontext úlohy se musí lišit.

**Řešení:**

Kód provádí simulaci hodu mincí pomocí metody Monte Carlo. Nejprve je nastaven počet hodů mincí na velké číslo, například 1 000 000. Poté kód generuje pseudonáhodná čísla reprezentující výsledky hodů mincí (0 pro hlavu a 1 pro pannu) a počítá počet hlav a pann. Na závěr jsou vypočítány pravděpodobnosti hlavy a panny dělením počtů hlav a pann celkovým počtem hodů. Tímto způsobem můžeme získat odhad pravděpodobností bez potřeby znát matematické vzorce nebo analytické řešení problému.

# 8. Derivace funkce jedné proměnné

**Zadání:**

Numerická derivace je velice krátké téma. V hodinách jste se dozvěděli o nejvyužívanějších typech numerické derivace (dopředná, zpětná, centrální). Jedno z neřešených témat na hodinách byl problém volby kroku. V praxi je vhodné mít krok dynamicky nastavitelný. Algoritmům tohoto typu se říká derivace s adaptabilním krokem. Cílem tohoto zadání je napsat program, který provede numerickou derivaci s adaptabilním krokem pro vámi vybranou funkci. Proveďte srovnání se statickým krokem a analytickým řešením.

**Řešení:**

Tento kód implementuje numerickou derivaci s adaptabilním krokem pro zadanou funkci. Využívá různé metody numerické derivace, jako je dopředná, zpětná a centrální diference, pro aproximaci derivace funkce. Adaptabilní derivace postupně zmenšuje krok a porovnává odchylku mezi derivacemi, aby dosáhla přesnějších výsledků. Testovací kód srovnává výsledky adaptivního kroku s analytickým řešením derivace, což ukazuje přesnost numerických aproximací.

# 

# 9. Integrace funkce jedné proměnné

**Zadání:**

V oblasti přírodních a sociálních věd je velice důležitým pojmem integrál, který představuje funkci součtů malých změn (počet nakažených covidem za čas, hustota monomerů daného typu při posouvání se v řetízku polymeru, aj.). Integraci lze provádět pro velmi jednoduché funkce prostou Riemannovým součtem, avšak pro složitější funkce je nutné využít pokročilé metody. Vaším úkolem je vybrat si 3 různorodé funkce (polynom, harmonická funkce, logaritmus/exponenciála) a vypočíst určitý integrál na dané funkci od nějakého počátku do nějakého konečného bodu. Porovnejte, jak si každá z metod poradila s vámi vybranou funkcí na základě přesnosti vůči analytickému řešení.

**Řešení:**

Kód implementuje výpočet určitého integrálu pomocí čtyř různých numerických metod (trapezoidní pravidlo, Simpsonovo pravidlo, Rombergova metoda a metoda kvadratury). Pro každou z těchto metod je definována samostatná funkce, která přijímá integrandu, dolní a horní mez integrace a další parametry potřebné pro výpočet. Funkce jsou poté volány pro tři různorodé funkce a výsledky jsou vypsány.

# 10. Řešení obyčejných diferenciálních rovnic

**Zadání:**

Diferenciální rovnice představují jeden z nejdůležitějších nástrojů každého přírodovědně vzdělaného člověka pro modelování jevů kolem nás. Vaším úkolem je vybrat si nějakou zajímavou soustavu diferenciálních rovnic, která nebyla zmíněna v sešitech z hodin a pomocí vhodné numerické metody je vyřešit. Řešením se rozumí vizualizace jejich průběhu a jiných zajímavých informací, které lze z rovnic odvodit. Proveďte také slovní okomentování toho, co lze z grafu o modelovaném procesu vyčíst.

**Řešení:**

V kódu je implementována Eulerova numerická metoda pro řešení Lotka-Volterra modelu, který popisuje interakci mezi druhy v ekosystému. Parametry rovnic ovlivňují chování systému, a graf zobrazuje změny počtu jedinců kořisti a dravců v průběhu času. Oscilace na grafu vyjadřují přirozenou interakci mezi druhy, a změny parametrů mohou mít vliv na stabilitu systému a výskyt různých jevů.